Deterministic Algorithm for Coded Cooperative Data Exchange*

Alex Sprintson[1], Parastoo Sadeghi[2], Graham Booker[1], and Salim El Rouayheb[3]

[1] Texas A&M University, College Station, Texas, USA
{spalex,gbooker}@tamu.edu
[2] Australian National University, ACT, Canberra, Australia
parastoo.sadeghi@anu.edu.au
[3] University of California at Berkeley, Berkeley, California USA
salim@eecs.berkeley.edu

Abstract. We consider the problem of cooperative data exchange in a group of wireless clients. In this problem each client initially holds a subset of packets and needs to obtain all packets held by other clients. Each client can broadcast its own packets or a combinations thereof to other clients via an error-free broadcast channel. Assuming that clients know which packets are available to other clients, our goal is to minimize the total number of transmissions needed to satisfy the demands of all clients. We present a deterministic algorithm that computes an optimal solution to this problem in polynomial time.

1 Introduction

In this paper, we consider the problem of cooperative data exchange between a group of wireless clients that share a common lossless broadcast channel. In this problem, a set of n packets $X = \{x_1, \dots, x_n\}$ needs to be delivered to k clients. Each client initially holds a subset X_i of packets in X and needs to obtain all packets held by other clients. Our goal is to design a communication scheme that enables all clients to obtain all packets with the minimum number of transmissions.

For example, consider the instance of the problem shown in Fig. 1(a) where there are three wireless clients that need to obtain three packets $x_1, x_2, x_3 \in \mathrm{GF}(2^m)$. Initially, the clients hold packets $\{x_2, x_3\}$, $\{x_1, x_3\}$ and $\{x_1, x_2\}$, respectively, i.e., each client is missing one packet. A simple cooperative scheme consists of three uncoded transmissions. However, this is not an optimal solution since the clients can send coded packets which satisfy demands of multiple clients. The number of transmissions for this example can be decreased to two by letting the first client broadcast $x_2 + x_3$ and the second client broadcast x_1 (see Fig. 1(b)).

In this paper, we present an algorithm that finds, in polynomial time, the optimal solution for the cooperative data exchange problem. In particular, the algorithm finds an encoding scheme that achieves the minimum number of transmissions over a small finite field.

* The work of Alex Sprintson was supported by NSF grant CNS-0954153. The work of Parastoo Sadeghi was supported under Australian Research Councils Discovery Projects funding scheme (project no. DP0984950).

X. Zhang and D. Qiao (Eds.): QShine 2010, LNICST 74, pp. 282–289, 2011.

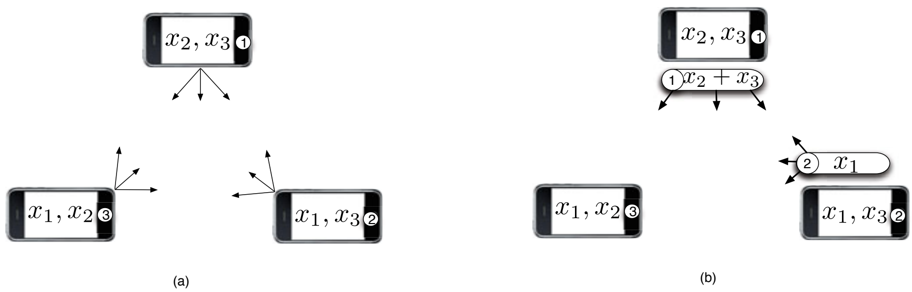

Fig. 1. Coded data exchange among three clients

Related Work

Cooperative communication at the physical, network and application layers has been the subject of extensive research in the past few years. Physical-layer user cooperation in the form of signal relaying has been shown to result in higher data rates, extended coverage, and robustness to link outages [9, 13, 15]. Network coding [1, 12] is another powerful technique that has been proposed to enhance achievable data rates, as well as other aspects, through packet-level encoding at intermediate nodes [11].

Direct information exchange problems were recently considered in [6] and [7] where it was assumed that the packets available at different network nodes follow a certain random distribution.

A closely-related coding problem with "subset" side information is the *Index Coding* problem [2, 3, 4, 5, 8] which was originally motivated by satellite broadcast applications with cashing clients. However, the Index Coding setup is centralized and non-cooperative with a single transmitter server holding all packets and passive clients having different demands. A related problem of set reconciliation between two or more similar sets was studied in [14].

When the clients have only information about certain neighboring nodes, and can communicate to a restricted number of them, our setting is related to that of gossip algorithms studied in the literature (see e.g., [16]).

In our previous work [17] we have presented an efficient randomized algorithm for the distributed data exchange problem and established several bounds on the minimum number of transmissions. In this paper, we extend our results by presenting an efficient deterministic algorithm for this problem.

2 Model

Consider a set of n packets $X = \{x_1, \ldots, x_n\}$ to be delivered to k clients belonging to the set $C = \{c_1, \ldots, c_k\}$. The packets are elements of a finite alphabet which will be assumed to be a finite field $\mathbb{F}_q$ throughout this paper. At the beginning, each client knows a subset of the packets denoted by $X_i \subseteq X$, while the clients collectively know all the packets in X, i.e., $\cup_{c_i \in C} X_i = X$. We denote by $\overline{X}_i = X \setminus X_i$ the set of packets missing at client c_i. We refer to X_i as the *has* set of client c_i and to $\overline{X}_i$ as its *wants* set. We assume that each client knows the indices of packets that are available to other clients.[1] Without loss of generality, we assume that each packet in X is needed by at least one client.

The clients exchange packets over a lossless broadcast channel with the purpose of making all packets in X available to all clients. The data is transferred in communication rounds, such that at round i one of the clients, say c_{t_i}, broadcasts a packet $p_i \in \mathbb{F}_q$ to the rest of the clients in C. Packet p_i may be one of the packets in X_{t_i}, or a combination of packets in X_{t_i} and the packets $\{p_1, \ldots, p_{i-1}\}$ previously transmitted over the channel. Our goal is to devise a scheme that enables each client $c_i \in C$ to obtain all packets in $\overline{X}_i$ while minimizing the total number of transmissions. We refer to the minimum number of transmissions required to satisfy all clients as OPT.

Our scheme uses linear coding over the field $\mathbb{F}_q$. As discussed in [17], linear codes are sufficient to achieve the minimum number of transmission in our problem. With linear coding, any packet p_i transmitted by the algorithm is a linear combination of the original packets in X, i.e.,

$$p_i = \sum_{x_j \in X} \gamma_i^j x_j,$$

where $\gamma_i^j \in \mathbb{F}_q$ are the *encoding coefficients* of p_i. We refer to the vector $\gamma_i = [\gamma_i^1, \gamma_i^2, \ldots, \gamma_i^n]$ as the *encoding vector* of p_i. The i-th *unit* encoding vector that corresponds to the original packet x_i is denoted by $u_i = [u_i^1, u_i^2, \ldots, u_i^n]$, where $u_i^i = 1$ and $u_i^j = 0$ for $i \neq j$. We also denote by U_i the set of unit vectors that correspond to packets in X_i.

A client c_i is said to have a *unique* packet x_j if $x_j \in X_i$ and $x_j \notin X_\ell$ for all $\ell \neq i$. A unique packet can be broadcast by the client holding it without any penalty in terms of optimality. Thus, without loss of generality, we assume that there are no unique packets in the system. Also, without loss of generality, we assume that all k clients initially have distinct packet sets.

3 Deterministic Algorithm

In this section, we present a deterministic algorithm for the data exchange problem. For clarity, we describe and analyze the behavior of the algorithm in terms of encoding vectors, rather than the original packets. That is, instead of saying that a packet $p_i = \sum_{x_j \in X} \gamma_i^j x_j$ has been transmitted, we say that we transmit the corresponding encoding vector $\gamma_i = [\gamma_i^1, \gamma_i^2, \ldots, \gamma_i^n]$.

3.1 Algorithm Description

Our algorithm operates over a finite field $\mathbb{F}_q$. The size q of $\mathbb{F}_q$ must be larger than $2k$, where k is the number of clients. For a client $c_j \in C$ we define by $\Gamma(c_j) = \mathtt{span}(U_j)$, i.e., $\Gamma(c_j)$ is the set of all possible encoding vectors in $\mathbb{F}_q^n$ that can be generated by client c_j. Then, each vector $\gamma_i \in \Gamma(c_j)$ can be written as

$$\gamma_i = \sum_{u_g \in U_j} \gamma_i^g u_g,$$

[1] This can be achieved by exchanging packet indices at the beginning of data exchange. The indices can also be piggybacked on the data packets to reduce overhead.

where each γ_i^g is an element of $\mathbb{F}_q$.

The algorithm executes in iterations. In each iteration we identify a client that will be transmitting at that round. The key idea is that at each round i we only determine the client c_{t_i} that will transmit a packet at that round, but not the encoding coefficients of the packet that will be transmitted. The encoding coefficients of each transmitted packets will be determined at the last stage of the algorithm.

More specifically, for each client $c_j \in C$ we maintain a counter b_j that specifies the number of the packets that will be transmitted by that client. Initially, $b_j = 0$ for all $c_j \in C$. Once we have determined that client c_{t_i} is transmitting a packet a round i, we increment the corresponding counter b_{t_i}. We denote by $B_i = (b_1, b_2, \dots, b_k)$ the vector that specifies the number of transmissions made by each client $c_j \in C$ at iteration i. We refer to B_i as a *counting vector.*

Definition 1. *We say that a set of vectors Γ fits $B_i = (b_1, b_2, \dots, b_k)$ if Γ can be partitioned into k disjoint subsets $\Gamma^1, \Gamma^2, \dots, \Gamma^k$, such that for each $\Gamma^j, 1 \le j \le k$ it holds that:*

1. $|\Gamma^j| = b_j$;
2. $\Gamma^j \subseteq \Gamma(c_j)$.

We also denote by $\mathcal{M}(B_i)$ the collection of all sets of vectors that fit B_i.

That is, the set Γ that fits B_i is a union of b_1 vectors from $\Gamma(c_1)$, b_2 vectors from $\Gamma(c_2)$, $\dots$, and b_k vectors from $\Gamma(c_k)$.

Definition 2. *Let B_i be a counting vector and let U_j be the set of encoding vectors available to client j. Let $\mathcal{M}(B_i)$ the collection of all sets of encoding vectors that fit B_i. Then, we define* $\texttt{Maxrank}(B_i, U_j)$ *as follows:*

$$\texttt{Maxrank}(B_i, U_j) = \max_{\Gamma \in \mathcal{M}(B_i)} \text{rank}(\Gamma \cup U_j).$$

For given B_i and U_j, the value of $\texttt{Maxrank}(B_i, U_j)$ can be efficiently computed in polynomial time. First, note, that there exits a set $\Gamma' \in \mathcal{M}(B_i)$ that maximizes the value of $\text{rank}(\Gamma \cup U_j)$ that only contains unit vectors u_g (corresponding to packets in X). Thus, we can compute $\texttt{Maxrank}(B_i, U_j)$ by constructing a bipartite graph $G(V_1, V_2, E)$, as depicted in Fig. 2. The nodes of V_1 correspond to packets in $X = \{x_1, \dots, x_n\}$. For each client $c_g \in C$, set V_2 contains b_g nodes, each node is connected to all nodes in V_1 that correspond to packets in X_g. In addition, set V_2 contains $|X_j| = |U_j|$ nodes, each node is connected to a corresponding packet in X_j. Note that each node in V_2 corresponds to a linear combination of packets in X, and our goal is to maximize the number of independent linear combinations. It is easy to verify that the value of $\texttt{Maxrank}(B_i, U_j)$ is equal to the maximum size of a matching in $G(V_1, V_2, E)$.

The formal description of the algorithm, referred to as *Deterministic Data Exchange (DDE)*, appears on Fig. 3.

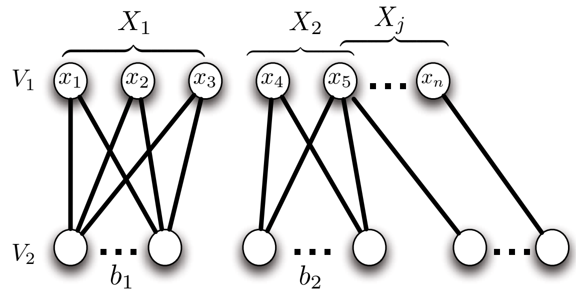

Fig. 2. The auxiliary graph $G = (V_1, V_2, E)$ used for computing the value of $\texttt{Maxrank}(B_i, U_j)$

Algorithm DDE $(C, \{U_j, c_j \in C, \mathbb{F}_q\})$

For *each* $c_j \in C$ ***do:***
  $b_j \leftarrow 0$
enddo
$B_0 \leftarrow (b_1, b_2, \ldots, b_k)$
$i \leftarrow 1$
while *there exists a client* $c_j \in C$ *for which it holds that* $\texttt{Maxrank}(B_{i-1}, U_j) < n$ ***do***
  Let c_{t_i} *be a client for which* $\texttt{Maxrank}(B_{i-1}, U_{t_i})$ *is maximum.*
  $b_{t_i} \leftarrow b_{t_i} + 1$
  $B_i = (b_1, b_2, \ldots, b_k)$
  $i \leftarrow i + 1$
endwhile
$\hat{i} \leftarrow i - 1$
Find a vector set $\hat{\Gamma} \in \mathcal{M}(B_{\hat{i}})$ *that satisfies* $\operatorname{rank}(\hat{\Gamma} \cup U_j) = n$ *for all* $c_j \in C$
return $\hat{i}$ *encoding vectors* $\gamma_1, \ldots, \gamma_{\hat{i}}$ *that correspond to elements in* $\hat{\Gamma}$

Fig. 3. Algorithm DDE

3.2 Algorithm Analysis

We proceed to analyze the correctness of the algorithm. Consider an iteration i of the algorithm. Recall that the vector $B_i = (b_1, b_2, \ldots, b_k)$ specifies the number transmissions made by each client $c_l \in C$ during iterations $1, \ldots, i$. Recall also that the collection $\mathcal{M}(B_i)$ includes all possible sets of encoding vectors that fit B_i.

Let OPT_i be the minimum number of additional rounds (i.e., starting from round $i + 1$) necessary to satisfy requests of all clients in C, provided that during iterations $1, \ldots, i$ the number of transmissions made by each client is consistent with B_i. We define OPT_0 as the optimal solution to the problem at hand, i.e., $OPT_0 = OPT$. Also, let $L_i = (l_1, l_2, \ldots, l_k)$ the number of additional transmissions that need to be done by the clients $c_j \in C$ to achieve optimum OPT_i. We refer to L_i as a *forward counting vector* at iteration i. Note that $OPT_i = \sum_{j=1}^{k} l_j$. Note also that the set L_i must satisfy that $\texttt{Maxrank}(B_i + L_i, U_j) = n$ for each client $c_j \in C$, where $B_i + L_i = (b_1 + l_1, b_2 + l_2, \ldots, b_k + l_k)$.

In order to prove the optimality of the algorithm it is sufficient to show that for each iteration $i = 1, 2, \dots$ it holds $OPT_i = OPT_{i-1} - 1$.

Theorem 1. *For each iteration i of the algorithm it holds that $OPT_i = OPT_{i-1} - 1$*

Proof: Consider iteration i of the algorithm. Let c_{t_i} be the client selected at that iteration. Let $B_{i-1} = (b_1, b_2, \dots, b_k)$ be the counting vector and $L_{i-1} = (l_1, l_2, \dots, l_k)$ be the forward counting vector at iteration $i - 1$.

First, consider the case in which $l_{t_i} > 0$. Let B_i be a vector formed from B_{i-1} by incrementing b_{t_i} by one. Let L_i be a vector formed from L_{i-1} be decrementing l_{t_i} by one. Note that $B_i + L_i = B_{i-1} + L_{i-1}$. Thus, after iteration i we need $OPT_{i-1} - 1$ transmissions to satisfy $\texttt{Maxrank}(B_{\hat{i}}, U_j) = n$ for each client $c_j \in C$. Hence it holds that $OPT_i = OPT_{i-1} - 1$.

Next, suppose that $l_{t_i} = 0$. Note that for each client $c_j \in C$ it holds that $\texttt{Maxrank}(B_{i-1} + L_{i-1}, U_j) = n$. Then, there exist vector set $Q_{i-1} \in \mathcal{M}(L_{i-1})$ and $\Gamma_{i-1} \in \mathcal{M}(B_{i-1})$ that satisfy, for each client $c_j \in C$,

$$\text{rank}(\Gamma_{i-1} \cup Q_{i-1} \cup U_j) = n. \tag{1}$$

Also, the definition of Maxrank implies that for each client $c_j \in C$ there exists a vector set Γ_{i-1} that satisfies

$$\text{rank}(\Gamma_{i-1} \cup U_j) = \texttt{Maxrank}(B_{i-1}, U_j). \tag{2}$$

By using the standard network coding techniques (see e.g., [12]) it can be shown that there exist sets of vectors $\Gamma_{i-1} \in \mathcal{M}(B_{i-1})$ and $Q_{i-1} \in \mathcal{M}(L_{i-1})$ that satisfy the conditions of both Equations (1) and (2) for all clients $c_j \in C$, provided that the field size q is larger than the number of constraints ($2k$).

Since client c_{t_i} has the largest value of $\texttt{Maxrank}(B_{i-1}, U_j)$ among all clients in $c_j \in C$, it holds that

$$n - \texttt{Maxrank}(B_{i-1}, U_{t_i}) = n - \text{rank}(\Gamma_{i-1} \cup U_{t_i}) \leq \sum_{j=1}^{k} l_j - 1 = OPT_{i-1} - 1.$$

This implies that there exists at least one vector $v \in Q_{i-1}$ such that the set $\Gamma_{i-1} \cup \{Q_{i-1} \setminus \{v\}\} \cup U_{t_i}$ is of rank n.

Let v be such a vector and let c_{i^*} be a client for which it holds that $v \in \Gamma(c_{i^*})$. We denote by $\tilde{Q}_{i-1} = Q_{i-1} \setminus \{v\}$. Note that for each client $c_j \in C \setminus \{c_{t_i}\}$ it holds the rank of vector set $S_j = \Gamma_{i-1} \cup \tilde{Q}_{i-1} \cup U_j$ is at least $n - 1$. Let C' be as subset of $C \setminus \{c_{t_i}\}$ such that for each $c_j \in C'$ it holds that $\text{rank}(S_j) = n - 1$.

Let c_j be a client in C' and let ζ_j be the normal vector to the span of S_j. Note that ζ_j is non-zero according to the definition of C'. Note that ζ_j can be written as

$$\zeta_j = \sum_{u_g \in U_{t_i}} \beta_g u_g + \sum_{u_g \in \overline{U}_{t_i}} \beta_g u_g,$$

where $\overline{U}_{t_i}$ is the set of unit encoding vectors that correspond to $\overline{X}_{t_i} = X \setminus X_{t_i}$.

Lemma 1. *There exists $u_g \in U_{t_i}$ such that $\beta_g \neq 0$.*

Proof: Suppose that it is not the case. Then, ζ_j can be expressed as $\zeta_j = \sum_{u_g \in \overline{U}_{t_i}} \beta_g u_g$. Then, ζ_j is a normal to $\mathtt{span}(U_{t_i})$. Since ζ_j is a normal to $\mathtt{span}(S_j)$ it is also normal to $\mathtt{span}(\Gamma_{i-1} \cup \tilde{Q}_{i-1})$. Thus, ζ_j is a normal to $\mathtt{span}(\Gamma_{i-1} \cup \tilde{Q}_{i-1} \cup U_{t_i})$ which contradicts the fact that $\mathrm{rank}\{\Gamma_{i-1} \cup \tilde{Q}_{i-1} \cup U_{t_i}\} = n$. ■

Let $\hat{\gamma}$ be a projection of ζ_j to $\mathtt{span}(U_{t_i})$, i.e., $\hat{\gamma} = \sum_{u_g \in U_{t_i}} \beta_g u_g$ and let $\Gamma_i = \Gamma_{i-1} \cup \{\hat{\gamma}\}$. Note that Lemma 1 implies that $\langle \hat{\gamma}, \zeta_j \rangle \neq 0$.

Lemma 2. *For each client $c_j \in C'$ it holds that*

$$\mathrm{rank}\{S_j \cup \{\hat{\gamma}\}\} = \mathrm{rank}\{\Gamma_i \cup \tilde{Q}_{i-1} \cup U_j\} = n.$$

Proof: By way of contradiction, suppose that $\mathrm{rank}\{S_j \cup \{\hat{\gamma}\}\} = n-1$. Then, vector $\hat{\gamma}$ belongs to $\mathtt{span}(S_j)$. However, this contradicts the fact that $\langle \hat{\gamma}, \zeta_j \rangle \neq 0$. ■

Lemma 2 implies that after iteration i the demands of all clients in C' are satisfied by transmitting vectors in $\tilde{Q}_{i-1}$, i.e., with $OPT_{i-1} - 1$ additional transmissions. The same holds for other clients in $C \setminus C'$, because for each client $c_j \in C \setminus C'$ it holds that $\mathrm{rank}\{S_j = \Gamma_{i-1} \cup \tilde{Q}_{i-1} \cup U_j\} = n$. Hence $OPT_i = OPT_{i-1} - 1$ and the lemma follows. ■

Lemma 3. *At Step 11 of Algorithm* DDE *it is possible to find a vector set $\hat{\Gamma} \in \mathcal{M}(B_{\hat{i}})$ that satisfies* $\mathrm{rank}(\hat{\Gamma} \cup U_j) = n$ *for all $c_j \in C$.*

Proof: At the end of iteration $\hat{i}$ of the algorithm, for each client $c_j \in C$ it holds that $\mathtt{Maxrank}(B_{\hat{i}}, U_j) = n$. The definition of `Maxrank` implies that for each client $c_j \in C$ there exists a vector set $\Gamma^j \in \mathcal{M}(B_{\hat{i}})$ such that $\mathrm{rank}(\Gamma^j \cup U_j) = n$. By using standard network coding techniques (see e.g., [10, 12]) we can find a set of vectors $\hat{\Gamma} \in \mathcal{M}(B_{\hat{i}})$ that satisfy the demands of all clients, provided that the field size q is larger than the number of constraints (k). ■

We summarize our results with the following theorem:

Theorem 2. *Algorithm* DDE *computes, in polynomial time, the optimal solution to the data exchange problem.*

Proof: Follows from Theorem 1 and the fact that OPT_0 is equal to the optimal solution OPT. ■

The computational complexity of the algorithm is comparable to that of the standard network coding algorithms such as due to Jaggi et al. [10].

4 Conclusions

We studied the problem of direct information exchange between a group of wireless clients with the goal to minimize the total number of transmissions between clients. We presented deterministic algorithm that provides an optimal solution in polynomial time.

There are many open problems for future research. One direction is to explore the two related issues of providing incentives to guarantee continued cooperation between clients and fairness to clients in terms of transmission load during data exchange. An additional interesting aspect of data exchange which can be considered is the energy cost associated with each transmission. This is of particular importance in networks with heterogeneous terminals that have different power consumption and battery options.

References

1. Ahlswede, R., Cai, N., Li, S.-Y.R., Yeung, R.W.: Network Information Flow. IEEE Trans. Inf. Theory 46(4), 1204–1216 (2000)
2. Alon, N., Hassidim, A., Lubetzky, E., Stav, U., Weinstein, A.: Broadcasting with Side Information. In: Proc. of the 49th Annual IEEE Symposium on Foundations of Computer Science (FOCS), pp. 823–832 (2008)
3. Bar-Yossef, Z., Birk, Y., Jayram, T.S., Kol, T.: Index Coding With Side Information. In: Proc. of the 47th Annual IEEE Symposium on Foundations of Computer Science (FOCS), pp. 197–206 (2006)
4. Birk, Y., Kol, T.: Coding-On-Demand by an Informed Source (ISCOD) for Efficient Broadcast of Different Supplemental Data to Caching Clients. IEEE Trans. Inf. Theory 52(6), 2825–2830 (2006)
5. Blasiak, A., Kleinberg, R., Lubetzky, E.: Index Coding via Linear Programming. IEEE Trans. Inf. Theory (April 2010) (submitted for publication), http://arxiv.org/abs/1004.1379
6. Courtade, T., Wesel, R.: Efficient Universal Recovery in Broadcast Networks. In: 48th Annual Allerton Conference on Communication, Control, and Computing, Monticello, IL, USA (September-October 2010)
7. Courtade, T., Wesel, R.D.: A Deterministic Approach to Rate-Compatible Fountain Communication. In: Information Theory and Applications Workshop (ITA), San Diego, CA, USA (January/February 2010)
8. El Rouayheb, S., Sprintson, A., Georghiades, C.: On the Index Coding Problem and Its Relation to Network Coding and Matroid Theory. IEEE Transactions on Information Theory 56(7), 3187–3195 (2010)
9. Hong, Y.-W., Huang, W.-J., Chiu, F.-H., Kuo, C.-C.J.: Cooperative Communications in Resource-Constrained Wireless Networks. IEEE Signal Process. Mag. 24(3), 47–57 (2007)
10. Jaggi, S., Sanders, P., Chou, P.A., Effros, M., Egner, S., Jain, K., Tolhuizen, L.: Polynomial Time Algorithms for Multicast Network Code Construction. IEEE Trans. Inf. Theory 51(6), 1973–1982 (2005)
11. Katti, S., Rahul, H., Hu, W., Katabi, D., Médard, M., Crowcroft, J.: XORs in The Air: Practical Wireless Network Coding. In: Proc. of ACM SIGCOMM, Pisa, Italy (2006)
12. Koetter, R., Médard, M.: An Algebraic Approach to Network Coding. IEEE/ACM Trans. on Networking 11(5), 782–795 (2003)
13. Laneman, J.N., Tse, D.N.C., Wornell, G.W.: Cooperative Diversity in Wireless Networks: Efficient Protocols and Outage Behavior. IEEE Trans. Inf. Theory 50(12), 3062–3080 (2004)
14. Minsky, Y., Trachtenberg, A., Zippel, R.: Set Reconciliation with Nearly Optimal Communication Complexity. IEEE Trans. Inf. Theory 49(9), 2213–2218 (2003)
15. Sendonaris, A., Erkip, E., Aazhang, B.: User Cooperation Diversity–Part I: System Description. IEEE Trans. Commun. 51(11), 1927–1938 (2003)
16. Shah, D.: Gossip algorithms. Foundations and Trends in Networking 3(1) (2007)
17. Sprintson, A., Sadeghi, P., Booker, G., El Rouayheb, S.: A Randomized Algorithm and Performance Bounds for Coded Cooperative Data Exchange. In: Proceedings of 2010 IEEE International Symposium on Information Theory (ISIT 2010), Austin, Texas, USA (June 2010)